High-Level Document:

Title: Quiz Application - High-Level Document

Introduction:

The Quiz Application is a web-based application that allows users to take a quiz and test their knowledge on various topics. The application presents multiple-choice questions to the user, provides options to select the answers, and evaluates the user's score at the end of the quiz.

Features:

a. Multiple-choice questions: The application displays questions with multiple answer options.

b. Answer selection: Users can select one option as their answer for each question.

c. Score calculation: The application calculates the user's score based on the correct answers.

d. Progress tracking: Users can see their progress in terms of the number of questions answered.

e. Result display: At the end of the quiz, the application shows the user's score and provides an option to play again.

Technologies Used:

a. HTML: Used for structuring the quiz layout and displaying the questions.

b. CSS: Used for styling the quiz interface.

c. JavaScript: Used for dynamically loading questions, handling user answers, and calculating the score.

Quiz Data:

The quiz questions and their corresponding options are stored in an array of objects in JavaScript. Each object contains the following properties:

question: The text of the question.

a, b, c, d: The answer options for the question.

ans: The correct answer option.

User Interaction Flow:

a. When the quiz is loaded, the first question is displayed along with the answer options.

b. The user selects one answer option for the current question.

c. The user clicks the "Submit" button to submit the answer.

d. The application checks if the selected answer is correct and increments the score if it is.

e. The application loads the next question and repeats the above steps until all questions are answered.

f. At the end of the quiz, the application displays the user's score and provides an option to play again.

Design and Styling:

The application has a responsive design that adapts to different screen sizes. It uses the Josefin Sans font from Google Fonts for a modern and clean look. The main quiz interface is contained within a centered div with a white background and a box shadow. The question and answer options are styled to be easily readable and visually appealing. The "Submit" button and the score area have distinct styles to stand out.

Conclusion:

The Quiz Application provides an interactive way for users to test their knowledge and have fun. It is implemented using HTML, CSS, and JavaScript, making it accessible on various devices and browsers. The application's user-friendly interface and scoring system enhance the user experience, encouraging users to engage with the quiz.

Low-Level Document:

Title: Quiz Application - Low-Level Document

HTML Structure:

a. The HTML structure consists of a main container div with the class "main-div".

b. Inside the main div, there is an inner div with the class "inner-div".

c. The inner div contains the following elements:

h2 with the class "question" for displaying the current question.

Unordered list (ul) for displaying the answer options.

List items (li) for each answer option.

Radio buttons (input type="radio") for selecting the answers.

Labels for associating the radio buttons with the answer options.

Button with the id "submit" for submitting the answer.

Div with the id "showScore" for displaying the score.

CSS Styling:

a. The CSS styling is done using the style.css file.

b. The main container div has a full viewport width and height and uses grid and flex properties for centering its content.

c. The inner div has a width of 40vw and padding for spacing.

d. The question, answer options, and other elements are styled with appropriate font sizes, margins, and colors.

e. The submit button and score area have distinct styles to differentiate them from other elements.

f. The hover effect is applied to the submit button for better interactivity.

JavaScript Functionality:

a. The JavaScript functionality is implemented in the script.js file.

b. The quiz questions and answer options are stored in the quizDB array of objects.

c. The loadQuestion function is responsible for loading the current question and answer options onto the page.

d. The getCheckAnswer function retrieves the selected answer from the user.

e. The submit event listener is attached to the submit button to handle user interactions.

f. On submit, the application checks the selected answer, increments the score if it is correct, and loads the next question.

g. When all questions are answered, the application displays the final score and provides an option to play again.

External Resources:

a. The application uses the Josefin Sans font from the Google Fonts library, imported using the @import rule.

b. The style.css and script.js files are linked to the HTML using the link and script tags, respectively.

Development Environment:

The application can be developed using any code editor or integrated development environment (IDE) that supports HTML, CSS, and JavaScript. The developer needs a web browser to run and test the application.

Deployment:

Once the development is complete, the Quiz Application can be deployed on a web server or hosting platform to make it accessible to users over the internet.

Testing:

It is essential to perform testing to ensure the Quiz Application functions correctly. The application should be tested for:

Loading questions and answer options correctly.

Selecting and submitting answers.

Calculating the score accurately.

Progressing to the next question after submitting an answer.

Displaying the final score and the option to play again.

Maintenance and Future Enhancements:

The Quiz Application may require maintenance and updates in the future, such as adding new quiz questions, improving the user interface, or enhancing the scoring system. Regular testing and bug fixing are necessary to ensure the application's smooth functioning.